Random program generation via lambda-terms

Dániel Leskó, Máté Tejfel

Department of Programming Languages and Compilers, Eötvös Loránd University

{ldani,matej}@elte.hu

In modern software technology, testing is still an essential and core part of the development cycle and will be in the coming years too, despite other slowly emerging approaches like proof assistants, formal verification approaches, etc. Thus the need for high-quality test data is still an important issue for every software development team. For "simpler" case there are a lot of mature, automated solution like QuickCheck [1], however, if the expected test data is quite complex and has to fulfill several internal invariants, like if your test data should be a program itself for testing compilers, source code transformation or analyzer tools, one can find much fewer options and off-the-shelf solutions.

A general, two-phase approach will be presented to tackle this problem. The first phase aims to generate random, typed λ-terms in closed, β-normal form. The generation is done in a top-down manner and guided with expected size and type information. The generated set of λ-terms has a uniform distribution over all correctly typed terms with a given size, ensured by a counting approach. [2]

The second phase translates these random λ-terms into high-level programs, based on a grammar describing the exact language constructs of the targeted high-level programming language. As the first phase of the generation is also guided by these language constructs (more precisely only with their types) the translation from λ-terms to high-level programs is always complete and successful.

This work was originally focused on functional programs and their random generation, but the concept – as a small example will show – could work for imperative languages as well. From the implementation point of view, this solution could be wrapped to form a QuickCheck generator, which would result in easy usability with test frameworks. [3]

References

[1] Claessen, K. & Hughes, J. *QuickCheck: a lightweight tool for random testing of Haskell programs*, Acm sigplan notices, Vol. 46., pp 53-64.

[2] Grygiel, K., & Lescanne, P. *Counting and generating lambda terms.* Journal of Functional Programming, 23(5), pp 594-628.

[3] Leskó, D., & Tejfel, M. *Testing framework for embedded languages* American Institute of Physics, pp 454-457.